

Elitism-based Immigrants for Ant Colony Optimization in Dynamic Environments: Adapting the Replacement Rate

Michalis Mavrovouniotis

Centre for Computational Intelligence (CCI),
School of Computer Science and Informatics,
De Montfort University, The Gateway,
Leicester, LE1 9BH, U.K.
Email: mmavrovouniotis@dmu.ac.uk

Shengxiang Yang

Centre for Computational Intelligence (CCI),
School of Computer Science and Informatics,
De Montfort University, The Gateway,
Leicester, LE1 9BH, U.K.
Email: syang@dmu.ac.uk

Abstract—The integration of immigrants schemes with ant colony optimization (ACO) algorithms showed promising results on different dynamic optimization problems (DOPs). The principle of integrating immigrants schemes within ACO is to introduce newly generated ants that will replace other ants in the current population. One of the most advanced immigrants schemes is the elitism-based immigrants scheme, where the best from the previous environment is used as the base to generate immigrants. So far, the replacement rate used for elitism-based immigrants in ACO is fixed. This paper examines the impact of the replacement rate on the performance of ACO algorithms with elitism-based immigrants. An adaptive replacement rate is proposed and compared with fixed and optimized replacement rates based on a series of DOPs. The experiments show that the adaptive scheme provides an automatic way to set a good value, although not the optimal one, for the replacement rate within ACO with elitism-based immigrants for DOPs.

I. INTRODUCTION

Ant colony optimization (ACO) algorithms have proved to be powerful methods to address challenging combinatorial optimization problems [4], [5], [7], [20]. Most problems addressed by ACO have stationary environments where the problems remain fixed during the optimization process. However, many real-world problems are dynamic optimization problems (DOPs) where the environment may change during the execution of the algorithm.

When a dynamic change occurs, it may take some time for ACO algorithms to adapt to the new environment [1], [14]. This is because ACO algorithms are designed especially to converge into a single optimum [2], [4]. Therefore, once the population converges into an optimum, then it will be difficult to escape from it and track the changing optimum. A simple strategy to address this issue is to restart the algorithm whenever a dynamic change occurs. However, this strategy may be inefficient, especially in cases where the changing environments are similar [10].

Immigrants schemes have proved that they are effective strategies to enhance the performance of ACO on different DOPs, including the dynamic travelling salesman problem (DTSP) [11], [15] and dynamic vehicle routing problem

(DVRP) [13]. The principle of immigrants schemes is to introduce newly generated ants to replace a small portion of ants in the executing population. Traditionally, immigrant ants are generated randomly and represent random solutions. Another advanced immigrants scheme is where the best solution of the previous environment is used as the base to generate elitism-based immigrants.

One important parameter when integrating immigrants schemes is the replacement rate. Usually, the replacement rate is set to a fixed value, e.g., a typical value is 0.2. However, the value of the replacement rate may vary under different DOPs. The impact of fixed and adaptive immigrants replacement rates was originally examined with evolutionary algorithms (EAs) for dynamic binary encoded problems [23]. In this paper, we focus on ACO algorithms integrated with the elitism-based immigrants scheme for dynamic routing problems, including the DTSP and DVRP. The dynamic benchmark generator proposed in [17] is used to generate a series of dynamic test cases and experiments are systematically conducted. The experiments show that the elitism-based immigrants ACO (EIACO) with an adaptive replacement rate outperforms the EIACO with a fixed replacement rate in most dynamic test cases.

The rest of the paper is organized as follows. Section II describes the DTSP and DVRP generated by the benchmark generator. Section III describes the EIACO algorithm used for the experiments. Section IV describes the adaptive mechanism for the immigrants replacement rate. Section V gives the experimental results. Finally, Section VI concludes this paper with discussion on future work.

II. DYNAMIC TEST ENVIRONMENTS

A. Routing Problems

A routing problem can be presented using a weighted graph. Let $G = (N, A)$ be a weighted graph where $N = \{1, \dots, n\}$ is a set of n nodes and $A = \{(i, j) : i \neq j\}$ is a set of arcs. Each node i has a location defined by (x, y) and each arc (i, j) is associated with a non-negative distance d_{ij} . Usually,

the distance matrix of a problem instance is defined as $\vec{D} = (d_{ij})_{n \times n}$.

1) *Travelling Salesman Problem (TSP)*: The TSP can be described as follows: given a collection of cities, we need to find the shortest path that starts from one city and visits each of the other cities once and only once before returning to the starting city.

Formally, the TSP is defined as follows. Let ψ_{ij} denote the binary decision variables defined as:

$$\psi_{ij} = \begin{cases} 1, & \text{if } (i, j) \text{ is covered in the tour,} \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

where $\psi_{ij} \in \{0, 1\}$. Then, the objective of the TSP is defined as:

$$f(x) = \min \sum_{i=1}^n \sum_{j=1}^n d_{ij} \psi_{ij}, \quad (2)$$

where n is the number of cities and d_{ij} is the distance between cities i and j .

2) *Vehicle Routing Problem (VRP)*: The basic VRP can be described as follows: given a number of vehicles with a fixed capacity, we need to satisfy the demand of all the customers, starting from and finishing at the depot, i.e., $\{0\}$. Hence, a VRP without the capacity constraint or with one vehicle can be seen as a TSP described above.

Formally, the VRP is defined as follows. Let ψ_{ij}^k denote the binary decision variables defined as:

$$\psi_{ij}^k = \begin{cases} 1, & \text{if } (i, j) \text{ is covered by vehicle } k, \\ 0, & \text{otherwise,} \end{cases} \quad (3)$$

where $\psi_{ij}^k \in \{0, 1\}$ for the k -th vehicle. Then, the objective of the VRP is defined as:

$$f(x) = \min \sum_{i=0}^n \sum_{j=0}^n d_{ij} \sum_{k=1}^u \psi_{ij}^k, \quad (4)$$

subject to:

$$\sum_{i=0}^n q_i \sum_{j=0}^n \psi_{ij}^k \leq Q^k, \forall k \in \{1, \dots, u\}, \quad (5)$$

$$\sum_{i=0}^n \psi_{il}^k - \sum_{j=0}^n \psi_{lj}^k = 0, \forall k \in \{1, \dots, u\}, l \in \{1, \dots, n\}, \quad (6)$$

$$\sum_{i=1}^n \psi_{i0}^k \leq 1, \forall k \in \{1, \dots, u\}, \quad (7)$$

$$\sum_{j=1}^n \psi_{0j}^k \leq 1, \forall k \in \{1, \dots, u\}, \quad (8)$$

where Eq. (5) is the capacity constraint that ensures that no vehicle can be overloaded; Eq. (6) ensures that if a vehicle arrives at a customer, it must also leave the customer; Eq. (7) and Eq. (8) ensure that each vehicle is not scheduled more than once.

B. Dynamic Benchmark Generators

Over the years, several dynamic benchmark generators have been proposed for the TSP and VRP that tend to model a real-world scenario, such as the DTSP with traffic factors [6], [15], [18], the DTSP with exchangeable cities [8], [9], the DVRP with dynamic demands [19] and the DVRP with traffic factors [13]. These benchmark generators modify the fitness landscape, whenever a dynamic change occurs, and cause the optimum value to change.

In this paper, the dynamic benchmark generator for permutation-encoded problems (DBGP) proposed in [17] is used, which can convert any stationary permutation-encoded benchmark problem instance to a DOP. The fitness landscape is not modified with DBGP, and thus, the optimum value remains the same (if known). This is because DBGP shifts the population of the algorithm to search to a new location in the fitness landscape. The main advantage of using the DBGP rather than the other generators is that one can observe how close to the optimum an algorithm can perform when a dynamic change occurs.

C. Constructing Dynamic Test Environments

Considering the TSP and VRP descriptions, each object, i.e., city or customer, $i \in N$ has a location defined by (x, y) and each link $(i, j) \in A$ is associated with a non-negative distance d_{ij} . Typically, the distance matrix of a problem instance is defined as $\vec{D} = (d_{ij})_{n \times n}$. DBGP generates the dynamic case as follows.

Every f iterations a random vector $\vec{V}(T)$ is generated that contains exactly $m \times n$ objects, where $T = \lceil t/f \rceil$ is the index of the period of change, t is the iteration count of the algorithm, f determines the frequency of change, where n is the size of the problem instance, and m determines the magnitude of change. More precisely, $m \in [0.0, 1.0]$ defines the degree of change, in which only the first $m \times n$ of $\vec{V}(T)$ object locations are swapped. Then, a randomly re-ordered vector $\vec{U}(T)$ is generated that contains the objects of $\vec{V}(T)$. Therefore, exactly $m \times n$ pairwise swaps are performed in \vec{D} using the two random vectors $(\vec{V}(T) \otimes \vec{U}(T))$, where “ \otimes ” denotes the swap operator.

III. ELITISM-BASED IMMIGRANTS SCHEME FOR DOPS

A. Description of ACO

ACO algorithms were initially developed to address stationary routing problems [4]. Within ACO, a population of μ ants construct feasible solutions on their forward mode, and update pheromone trails on their backward mode every iteration. Hence, after a few iterations high concentrations of pheromone will be generated into a trail that corresponds to a solution of the routing problem, e.g., the TSP or VRP.

Each ant k constructs a feasible solution using a probabilistic rule to move from object i to object j as follows:

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in \mathcal{N}_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta}, \text{ if } j \in \mathcal{N}_i^k, \quad (9)$$

where τ_{ij} is the existing pheromone trail between objects i and j , $\eta_{ij} = 1/d_{ij}$ is the heuristic information available a priori, d_{ij} is the distance between objects i and j and \mathcal{N}_i^k denotes the objects that ant k is allowed to select from object i . Note that the main difference when constructing a TSP or a VRP solution lies in the generation of \mathcal{N}_i^k . For the TSP, \mathcal{N}_i^k is generated by the unvisited cities incident to city i , whereas for the VRP, \mathcal{N}_i^k is generated by the unvisited customers incident to customer i including the depot.

B. ACO for DOPs

ACO algorithms can transfer knowledge via the pheromone trails generated from the previous environment when the changing environments are similar. If the dynamic change is severe, the pheromone trails of the previous environment may misguide the population to search into areas far from the optimum; otherwise, the pheromone trails of the previous environment may provide knowledge to speed up the optimization to the new environment.

ACO algorithms suffer from the stagnation behaviour when addressing DOPs. For example, once the population converges quickly into an optimum, then it is difficult for the population to escape from the old optimum in order to adapt to the new optimum when an environmental change occurs. More precisely, the pheromone trails generated around the optimum before the dynamic change may misguide ants not to track the moving optimum. Therefore, ACO algorithms may need some time to adapt to dynamic changes because of the stagnation behaviour [14].

Over the years, several strategies have been proposed to address this issue and enhance the performance of ACO algorithms for DOPs, including maintaining diversity strategies [6], [8], [14], increasing diversity via immigrants [11], [13], [15], memory-based schemes [9], multi-colony schemes [18] and memetic algorithms [12], [16]. The integration of immigrants schemes within ACO algorithms has shown to be able to enhance the performance of ACO on the DTSP [15] and DVRP [13].

C. Elitism-based Immigrants ACO

Immigrants schemes mainly differ in the way immigrant ants are generated. Traditionally, immigrants are generated randomly and represent random solutions of the routing problem. However, too much randomization may disturb the optimization process. Elitism-based immigrants are generated via transferring knowledge from the previous environment, and hence the diversity generated is controlled. Elitism-based immigrants have shown more consistent performance when integrated with EAs for binary dynamic functions [22] and the dynamic shortest path problem in mobile ad hoc networks [3], with a CHC-based algorithm for the DTSP [21] and with ACO on different routing problems [13], [15].

In this paper, we focus on the integration of elitism-based immigrants with an ACO algorithm, known as EIACO. Within EIACO, the best ant from the previous environment is used as the base to generate $r_i \times k_s$ elitism-based immigrants to

replace the worst ants in $k_{short}(t)$. More precisely, an elitism-based immigrant is generated by swapping objects with a p_m^i probability. For the TSP, all the objects are allowed to be swapped whereas for the VRP only the objects that belong to the same vehicle route are allowed.

The solution construction of EIACO is the same as in Eq. (9). Every iteration t , the pheromone table is associated with a short-term memory of size k_s , denoted $k_{short}(t)$ in this paper, and any change to $k_{short}(t+1)$ causes an update to the pheromone table. The k_s best ants from the current iteration t replace the ants in $k_{short}(t-1)$.

When the worst ants are replaced by elitism-based immigrant ants, the pheromone trails of each k -th worst ant are removed, as follows:

$$\tau_{ij} \leftarrow \tau_{ij} - \Delta\tau_{ij}^k, \forall (i, j) \in T^k, \quad (10)$$

where T^k represents the tour of ant k and $\Delta\tau_{ij}^k = (\tau_{max} - \tau_0)/K_s$, where τ_{max} and τ_0 denote the maximum and initial pheromone values, respectively. Furthermore, the pheromone trails of each k -th immigrant ant are added, as follows:

$$\tau_{ij} \leftarrow \tau_{ij} + \Delta\tau_{ij}^k, \forall (i, j) \in T^k, \quad (11)$$

where $\Delta\tau_{ij}^k$ and T^k are as defined in Eq. (10).

IV. ADAPTING THE REPLACEMENT RATE

A. The Role of the Replacement Rate

The replacement rate is an important parameter when immigrants schemes are integrated with ACO to address DOPs. As an evidence, the experiments with different fixed replacement rates with EIACO on different DTSPs with traffic factors showed that the performance of EIACO depends significantly on the setting value of the replacement rate [15].

Since elitism-based immigrants transfer knowledge from previous environments, it is natural to set the replacement rate to a good value in order to have a positive effect on the performance of ACO for DOPs. It is straightforward that the optimization process cannot be disturbed by too much randomization since elitism-based immigrants generate diversity via transferring knowledge. However, too much knowledge transferred may degrade the performance for DOPs because it may start the re-optimization process near a local optimum and get stuck to it.

A fixed replacement rate may not be the best choice, because at different stages of the optimization process and under different DOPs the most appropriate replacement rate may vary. An adaptive replacement rate was initially investigated with EAs and showed promising results on binary-encoded DOPs [23]. In this paper, we investigate the adaptive replacement rate with ACO algorithms on dynamic routing problems.

B. Evaluating the Effect of Immigrants

In order to evaluate the effect of elitism-based immigrants, we use the adaptation method proposed in [23], where the tour cost of the generated immigrants is compared with the median tour cost of the actual population in every iteration t .

Let $\xi(t)$ denote the effect the elitism-based immigrants generated at iteration t , which can be defined as follows:

$$\xi(t) = \frac{n^i \{T^i(t) \leq T^{med}(t)\}}{n^i} \quad (12)$$

where $n^i = r_i \times k_s$ is the number of immigrants generated, $T^i(t)$ is the solution quality of the i -th immigrant and $T^{med}(t)$ is the median solution quality of the current ant population. This measurement gives a ratio of the immigrants that may help to guide the population into promising areas. Therefore, according to the number of elitism-based immigrants that have the solution quality close to the best solution of the previous environment the effect $\xi(t)$ increases.

C. Adaptive Rules for the Replacement Rate

The effect calculated in Eq. (12) is considered in order to adapt the replacement rate of immigrants. More precisely, if the effect is positive, then the replacement rate should be increased; otherwise, it should be decreased. Let $r_i(t)$ denote the replacement rate of immigrants at iteration t . Then, using the current effect of immigrants $\xi(t)$, the new replacement rate $r_i(t+1)$ can be updated as follows:

$$r_i(t+1) = \begin{cases} r_i(t) + 0.1, & \text{if } \xi(t) > \theta, \\ r_i(t) - 0.1, & \text{if } \xi(t) < \theta, \\ r_i(t), & \text{otherwise.} \end{cases} \quad (13)$$

where θ is a predefined threshold that defines whether there is a positive or negative effect and $r_i(t)$ is bounded to the interval $[0.0, 1.0]$. It can be observed that this adaptation method introduces a new parameter, i.e., θ . However, we will show later on that it is not a sensitive parameter comparing with the replacement rate, r_i , in terms of performance. Another possible (non-parametric) adaptation way is to adjust the replacement rate probabilistically, which deserves future investigation.

V. EXPERIMENTAL STUDY

A. Experimental setup

The EIACO algorithm is used¹ to investigate whether an adaptive or fixed replacement rate of immigrants schemes performs better for DOPs. The EIACO with an adaptive replacement rate is denoted as AR-EIACO and the EIACO with a fixed replacement rate is denoted as FR-EIACO.

All common algorithmic parameters were set as follows: $\alpha = 1$, $\beta = 5$, the population size $\mu = 30$, k_s was set to 10 and the elitism-based immigrant mutation probability p_m^i was set to 0.01. In order to investigate r_i 's effect in terms of performance, the replacement rate of FR-EIACO was set in $r_i \in \{0.0, 0.2, 0.4, 0.6, 0.8\}$ (see Section V-B) and the threshold θ of AR-EIACO was set in $\theta \in$

¹Random immigrant ACO is not considered because according to the adaptive rule, the effect in Eq. (12) will be zero for almost all of the times. This is because the solution quality of a random immigrant, i.e., a random solution, is more likely to be worse than the median solution quality of the population.

TABLE I
DYNAMIC OPTIMIZATION PROBLEM INDICES USED IN THE EXPERIMENTS

DOP Settings	$m = 0.1$	$m = 0.25$	$m = 0.5$	$m = 0.75$
$f = 10$	1	2	3	4
$f = 100$	5	6	7	8

$\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ (see Section V-C). For AR-EIACO, the initial value $r_i(0)$ was set to 0.5.

DOPs are generated from three TSP stationary benchmark instances obtained from TSPLIB² and three VRP stationary benchmark instances obtained from VRPLIB³ using the DBGp generator with f set to 10 and 100 indicating quickly and slowly changing environments, respectively, and m set to 0.1, 0.25, 0.5 and 0.75, indicating slowly, to medium, to severely changing environments, respectively. Totally, a series of 8 DOPs are constructed from each stationary instance for both DTSPs and DVRPs. The index of each DOP is described in Table I.

For each ACO algorithm on a DOP, 30 independent runs were executed on the same set of random seeds. For each run, 1000 iterations were allowed and the best so far ant after a dynamic change was recorded every iteration. The overall offline performance [10] of an ACO on a DOP is defined as follows:

$$\bar{P}_{OFF} = \frac{1}{G} \sum_{i=1}^G \left(\frac{1}{N} \sum_{j=1}^N P_{ij}^* \right), \quad (14)$$

where P_{ij}^* defines the tour cost of the best ant of iteration i of run j .

B. Effect of the Replacement Rate r_i

The offline performance results of FR-EIACO with different immigrants' replacement rates are plotted in Fig. 1 and Fig. 2 for DOPs with $f = 10$ and $f = 100$, respectively. It can be observed that different replacement rates achieve better performance on different DOPs. This support our claim that the replacement rate is a sensitive parameter and affects the performance of EIACO.

When $r_i > 0.0$, the performance of FR-EIACO is better than when $r_i = 0.0$ in almost all test cases, except on some DTSPs with $f = 10$. This is because elitism-based immigrants often guide the process towards promising areas and speed up re-optimization. However, in cases where the dynamic changes occur quickly, there is a high risk to misguide the search process. This may be due to the limited time available to gain knowledge in quickly changing environments.

Another observation is that higher replacement rates, e.g., $r_i > 0.2$, improve the performance of FR-EIACO on most DVRPs and degrade the performance on most DTSPs. This is natural because the swaps on the objects to generate the elitism-based immigrants on the DTSP occur in a larger

²<http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>

³<http://neo.lcc.uma.es/vrp/>

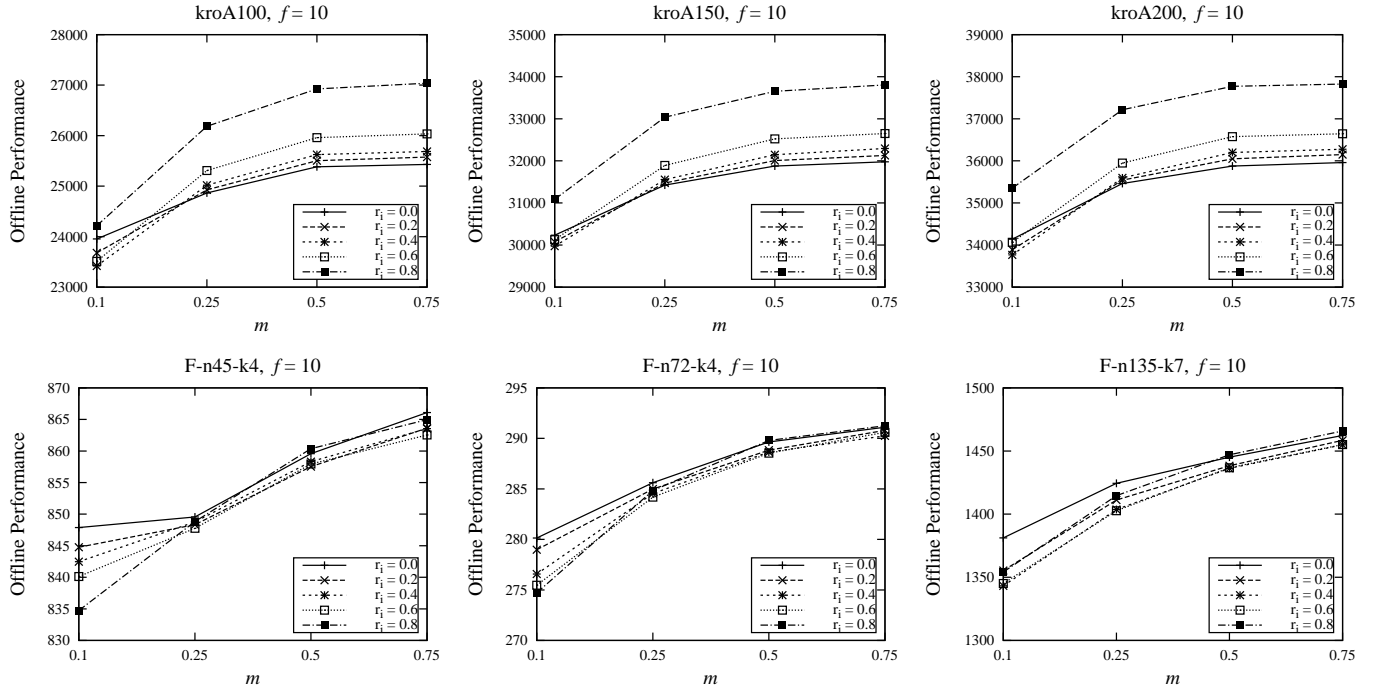


Fig. 1. Offline performance of FR-EIACO with different replacement rate values in quickly changing DOPs.

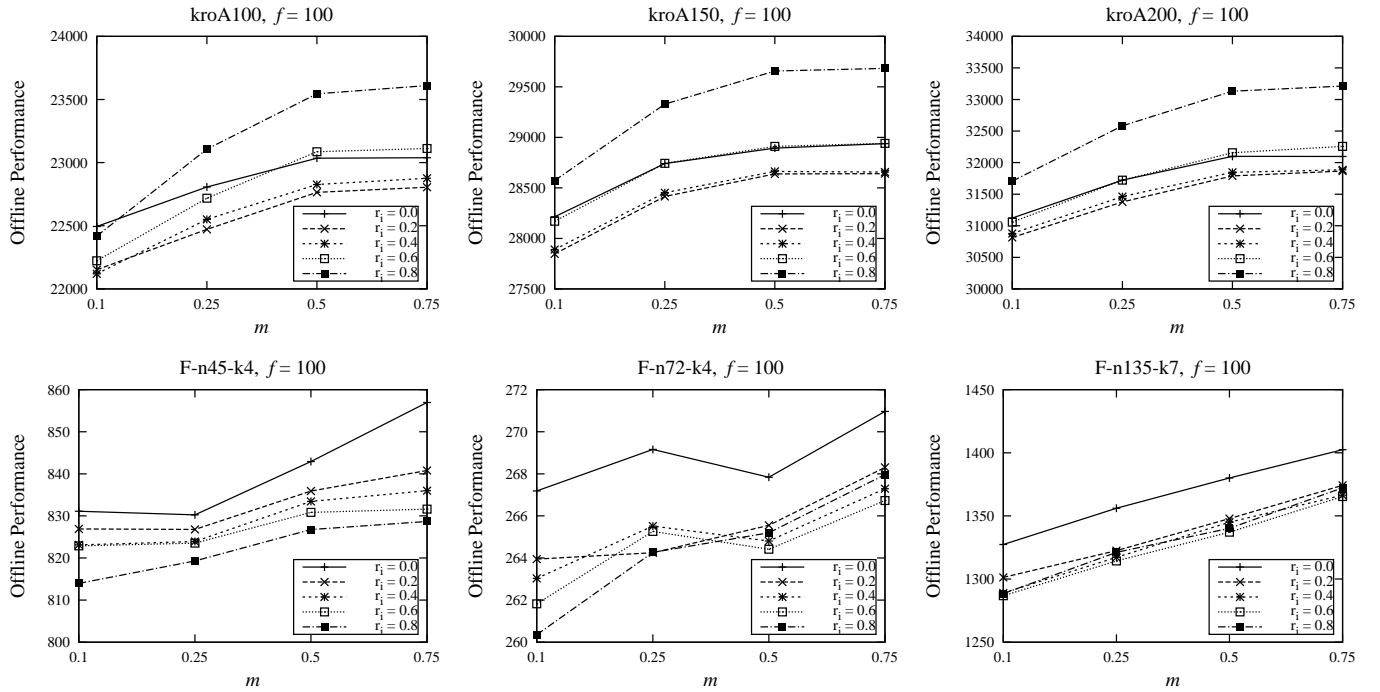


Fig. 2. Offline performance of FR-EIACO with different replacement rate values in slowly changing DOPs.

dimension, e.g., kroA200 has one large block of objects of size 200 where F-n135-k7 has many small blocks of objects and the size depends on the number of vehicles. In this way, even a single swap on the best solution of the previous environment in the DTSP has a higher probability to cause a big change to

the newly generated immigrant and misguide the searching in the new environment.

C. Effect of the Threshold θ

The offline performance results of AR-EIACO with different threshold values are plotted in Fig. 3 for all DOPs. The

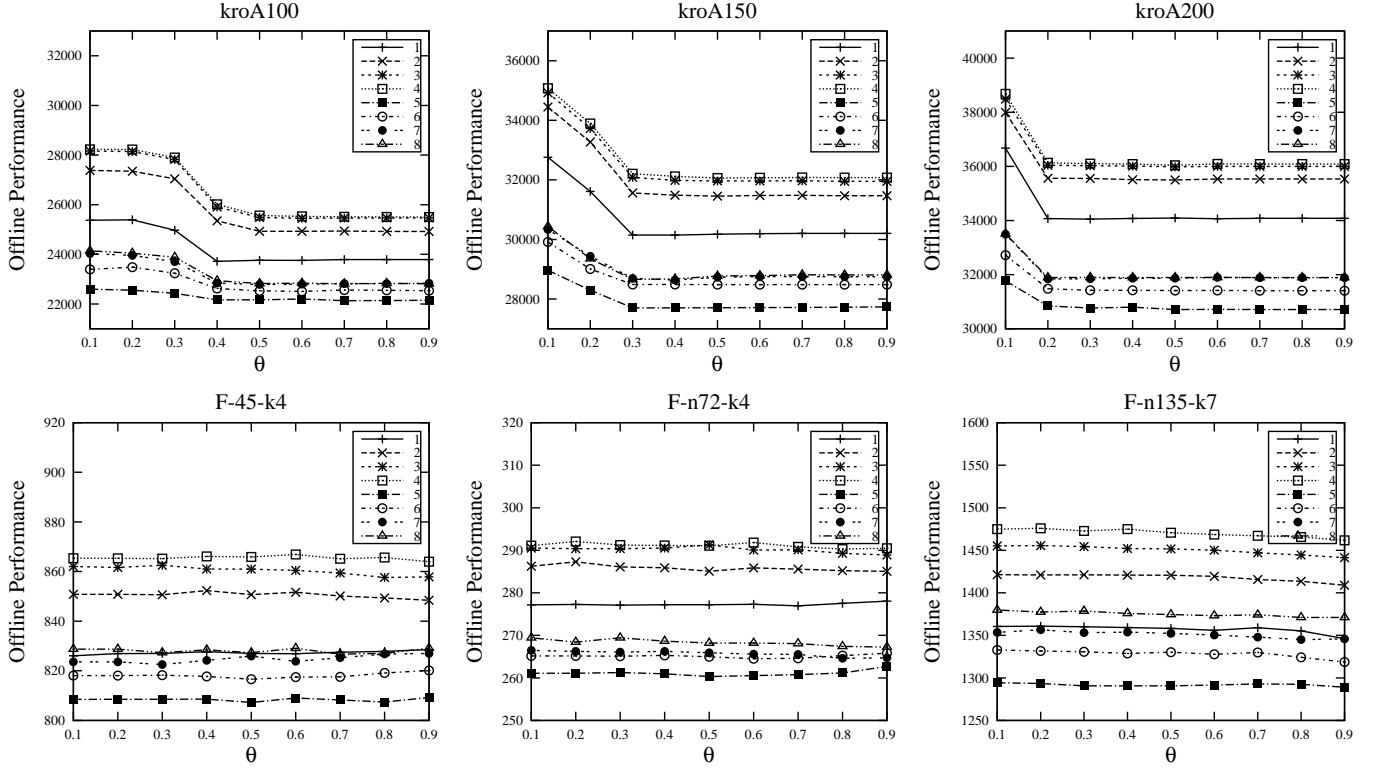


Fig. 3. Offline performance of AR-EIACO with different threshold values in DOPs on different environmental indices.

meaning of θ in Eq. 13 is to define whether the effect measured in Eq. 12 is positive or negative. Hence, it is natural that when θ is set to a small value the immigrants are more likely to be considered as having a positive effect, whereas when θ is set to a large value the immigrants are more likely to be considered as having a negative effect.

It can be observed that the parameter θ is more sensitive on the DTSP because the performance of AR-EIACO is degraded when $\theta < 0.5$. It was observed before that the elitism-based immigrants generated for the DVRP are more effective than the elitism-based immigrants generated for the DTSP because of the dimension that the object's swaps occur; see Section V-B. Therefore, a θ value that works in favour of a positive effect, i.e., a small value, will degrade the performance of AR-EIACO. In case that the elitism-based immigrants are always effective (even when r_i is large), e.g., on the DVRP case, any θ value will have similar performance on the AR-EIACO.

Another observation is that when $\theta \geq 0.5$, the performance of AR-EIACO is nearly the same for all DOPs. This shows that the threshold is not a sensitive parameter, comparing with the r_i parameter, and a value $\theta \in [0.5, 0.9]$ can achieve satisfactory performance for all DOPs, either DTSPs or DVRPs.

D. Adaptive versus Fixed and Optimized Replacement Rate

The experimental results regarding the offline performance of the AR-EIACO algorithm with $\theta = 0.7$ compared with the FR-EIACO with the commonly used $r_i = 0.2$ replacement rate value, denoted as FR-EIACO, and the best replacement

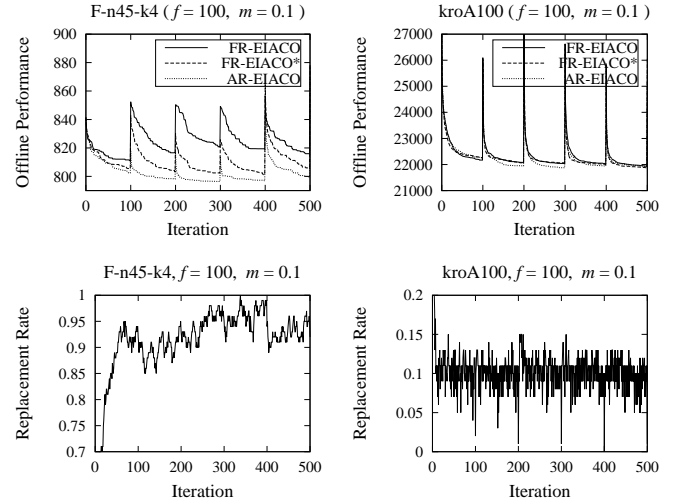


Fig. 4. Dynamic behaviour of ACO algorithms in terms of the offline performance with the corresponding replacement rate values of AR-EIACO with $\theta = 0.7$ on DOPs with $f = 100$ and $m = 0.1$ averaged over 30 runs for the first five environments.

value, denoted as FR-EIACO*, for all DOPs are presented in Table II. The corresponding statistical results are presented in Table III, where Kruskal-Wallis tests were applied followed by posthoc paired comparisons using Mann-Whitney tests with the Bonferroni correction. In Table III, the results are shown as “+”, “−” and “~” when the first algorithm is

TABLE II
EXPERIMENTAL RESULTS REGARDING THE OFFLINE PERFORMANCE OF EIACO WITH ADAPTIVE VERSUS FIXED IMMIGRANTS REPLACEMENT RATE

Travelling Salesman Problem												
Algorithms & DOPs	kroA100(Optimum=21282)				kroA150(Optimum=26524)				kroA200(Optimum=29368)			
$f = 10, m \Rightarrow$	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
AR-EIACO	23791	24937	25465	25514	30203	31479	31968	32081	34080	35531	35998	36086
FR-EIACO	23677	24926	25503	25575	30077	31473	32006	32130	33887	35541	36048	36148
FR-EIACO*	23418	24865	25384	25429	29973	31423	31877	31978	33766	35461	35876	35959
$f = 100, m \Rightarrow$	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
AR-EIACO	22131	22564	22819	22811	27716	28486	28636	28674	30711	31402	31888	31894
FR-EIACO	22152	22471	22764	22805	27846	28415	28638	28641	30814	31377	31788	31866
FR-EIACO*	22119	22471	22764	22805	27846	28415	28638	28641	30814	31377	31788	31866

Vehicle Routing Problem												
Algorithms & DOPs	F-n45-k4(Optimum=724)				F-n72-k4(Optimum=237)				F-n135-k7(Optimum=1162)			
$f = 10, m \Rightarrow$	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
AR-EIACO	827.5	848.4	857.8	864.0	276.9	285.2	289.2	291.2	1346.1	1408.9	1441.3	1461.6
FR-EIACO	844.7	848.3	857.5	863.6	278.9	284.9	288.8	290.7	1355.3	1411.1	1438.3	1458.4
FR-EIACO*	834.7	847.7	857.5	862.5	274.7	284.1	288.5	290.2	1343.1	1402.6	1436.2	1454.9
$f = 100, m \Rightarrow$	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
AR-EIACO	808.2	817.5	825.2	826.7	260.7	264.5	264.6	267.2	1288.9	1318.7	1345.7	1371.0
FR-EIACO	826.8	826.7	835.9	840.7	263.9	266.1	265.5	268.3	1301.3	1322.5	1347.9	1374.2
FR-EIACO*	813.9	819.3	826.7	828.6	260.3	264.2	264.4	266.7	1286.7	1314.2	1337.1	1365.3

TABLE III
STATISTICAL RESULTS REGARDING THE OFFLINE PERFORMANCE OF EIACO WITH ADAPTIVE VERSUS FIXED IMMIGRANTS REPLACEMENT RATE

Travelling Salesman Problem												
Algorithms & DOPs	kroA100				kroA150				kroA200			
$f = 10, m \Rightarrow$	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
AR-EIACO \Leftrightarrow FR-EIACO	-	~	+	+	-	~	+	+	-	~	+	+
AR-EIACO \Leftrightarrow FR-EIACO*	-	-	-	-	-	-	-	-	-	-	-	-
FR-EIACO \Leftrightarrow FR-EIACO*	-	-	-	-	-	-	-	-	-	-	-	-
$f = 100, m \Rightarrow$	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
AR-EIACO \Leftrightarrow FR-EIACO	~	~	~	~	+	~	~	~	+	~	~	~
AR-EIACO \Leftrightarrow FR-EIACO*	~	~	~	~	+	~	~	~	+	~	~	~
FR-EIACO \Leftrightarrow FR-EIACO*	~	~	~	~	~	~	~	~	~	~	~	~

Vehicle Routing Problem												
Algorithms & DOPs	F-n45-k4				F-n72-k4				F-n135-k7			
$f = 10, m \Rightarrow$	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
AR-EIACO \Leftrightarrow FR-EIACO	+	~	~	~	+	-	-	-	+	~	-	-
AR-EIACO \Leftrightarrow FR-EIACO*	+	~	~	~	-	-	~	~	~	-	-	-
FR-EIACO \Leftrightarrow FR-EIACO*	-	~	~	~	-	-	-	-	-	-	-	-
$f = 100, m \Rightarrow$	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
AR-EIACO \Leftrightarrow FR-EIACO	+	+	+	+	+	~	+	+	+	~	~	~
AR-EIACO \Leftrightarrow FR-EIACO*	+	~	~	~	~	~	~	~	~	~	-	~
FR-EIACO \Leftrightarrow FR-EIACO*	-	-	-	-	-	-	-	-	-	~	-	-

significantly better than the second one, when the second algorithm is significantly better than the first one, and when the two algorithms are not significantly different, respectively. In Fig. 4, the offline performance, together with the replacement rate values of AR-EIACO, against the algorithmic iterations of the algorithms are plotted to better understand the behaviour of the ACO algorithms. From the experimental results, several observations can be drawn.

First, AR-EIACO significantly outperforms FR-EIACO on most DVRLPs, except when $f = 10$ with $m = 0.5$ and $m = 0.75$; see the comparisons of AR-EIACO \Leftrightarrow FR-EIACO in Table III. This is because the generated elitism-based immigrants maintain a good solution quality when the changing environments give enough time to the algorithm to converge into a good solution or when the changing environments are

similar to transfer knowledge. Therefore, they will have a positive effect and increase the replacement rate. This can be observed from Fig. 4, where AR-EIACO maintains a higher replacement rate after the first environment. Moreover, it can be observed that on the first environment the performance of all algorithms is similar, whereas on the next environments AR-EIACO performs significantly better.

Second, AR-EIACO significantly outperforms FR-EIACO on DTSPs with $f = 10$ and $m = 0.5$ and $m = 0.75$ whereas FR-EIACO significantly outperform AR-EIACO on DTSPs with $f = 10$ and $m = 0.1$. This is due to the reasons explained in Section V-B that for the DTSP, elitism-based immigrants may not maintain a good solution quality because of the larger blocks. Therefore, they will have a negative effect and decrease the replacement rate according to the adaptive

method in Eq. 12. A lower replacement rate works in favour of FR-EIACO which can be observed from Fig. 1, where on most DTSPs with $f = 10$ the EIACO with no immigrants, i.e., $r_i = 0.0$, has the best performance. Differently, AR-EIACO is comparable with FR-EIACO on most DTSPs with $f = 100$, except when $m = 0.1$. This is because the replacement rate, i.e., $r_i = 0.2$, of FR-EIACO on DTSPs with $f = 100$ is the optimized one, i.e., FR-EIACO*, on almost all cases, which can be observed from Table II. When $f = 100$ and $m = 0.1$ AR-EIACO significantly outperforms FR-EIACO and FR-EIACO* on most DTSPs because the dynamic change is small.

Third, FR-EIACO* significantly outperforms AR-EIACO on most DOPs (both DTSPs and DVRPs), except on some cases from F-n45-k4, kroA150 and kroA200 when $m = 0.1$; see the comparisons of AR-EIACO \Leftrightarrow FR-EIACO* in Table III. This is natural because the replacement rate of FR-EIACO* is obtained via fine tuning whereas for AR-EIACO starts from $r_i(0) = 0.5$ and it is adapted during the execution as presented in Fig. 4. The performance of AR-EIACO is still satisfactory because it is usually not significantly worse than FR-EIACO*, which can be observed from Tables II and III. Considering the time consuming task of optimizing the replacement rate for every DOP, it is worth using the adaptive replacement rate, i.e., AR-EIACO with $\theta \in [0.5, 0.9]$, even if the performance is slightly degraded.

VI. CONCLUSIONS AND FUTURE WORK

This paper investigates the impact of the replacement rate within ACO algorithms with immigrants schemes for DOPs. Experiments were conducted on ACO algorithms with the well-known elitism-based immigrants scheme with an adaptive, fixed or optimized replacement rate. From the experiments, several concluding remarks can be drawn.

First, the replacement rate is an important parameter for EIACO when addressing DOPs. Second, different replacement rate values perform well on different DOPs. Third, the adaptive replacement rate performs better than the fixed replacement rate on most DOPs. Fourth, the elitism-based immigrants improve the performance of ACO on almost all DOPs, especially on DVRPs. Finally, the optimized replacement rate outperforms the adaptive replacement rate on most DOPs. However, it may be better to use an automatic adaptive replacement rate rather than a manual fixed replacement rate. Moreover, the optimization of the replacement rate parameter for each DOP requires computation time. Hence, for the sake of convenience, it may be better to adopt the adaptive replacement rate and slightly sacrifice the performance of the algorithm for DOPs.

For future work it will be interesting to apply other ways, i.e., non parametric, to adapt the immigrants replacement rate. Another future work is to adapt other important parameters of immigrants, e.g., the immigrants mutation probability p_m^i in EIACO.

ACKNOWLEDGMENT

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) of UK under Grant EP/K001310/1.

REFERENCES

- [1] D. Angus and T. Hendtlass, "Ant colony optimization applied to dynamically changing problem," in *Developments in Applied Artificial Intelligence*, ser. LNAI, vol. 2358. Springer-Verlag, 2002, pp. 618–627.
- [2] E. Bonabeau, M. Dorigo, and G. Theraulaz, Eds., *Swarm Intelligence: From Natural to Artificial Systems*. New York: Oxford University Press, 1997.
- [3] H. Cheng and S. Yang, "Genetic algorithms with elitism-based immigrants for dynamic shortest path problem in mobile ad hoc networks," in *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*, 2009, pp. 3135–3140.
- [4] M. Dorigo, V. Maniezzo, and A. Colnori, "Ant system: Optimization by a colony of cooperating agents," *IEEE Transactions on System Man and Cybernetics-Part B: Cybernetics*, vol. 26, no. 1, pp. 29–41, 1996.
- [5] M. Dorigo and T. Stützle, Eds., *Ant colony optimization*. London, England: MIT Press, 2004.
- [6] C. Eyckelhof and M. Snoek, "Ant systems for a dynamic tsp," in *Proceedings of the 3rd International Workshop on Ant Algorithms*, ser. LNCS, M. Dorigo, G. D. Caro, and M. Sampels, Eds., vol. 2463. Springer-Verlag, 2002, pp. 88–99.
- [7] L. M. Gambardella, E. D. Taillard, and C. Agazzi, "Macs-vrptw: A multicolony ant colony system for vehicle routing problems with time windows," in *New Ideas in Optimization*, 1999, pp. 63–76.
- [8] M. Guntsch and M. Middendorf, "Pheromone modification strategies for ant algorithms applied to dynamic tsp," in *EvoWorkshops 2001: Applications of Evolutionary Computing*, ser. LNCS, vol. 2037. Springer-Verlag, 2001, pp. 213–222.
- [9] —, "Applying population based aco to dynamic optimization problems," in *Proceedings of the 3rd International Workshop on Ant Algorithms*, ser. LNCS, M. Dorigo, G. D. Caro, and M. Sampels, Eds., vol. 2463. Springer-Verlag, 2002, pp. 111–122.
- [10] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments - a survey," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 3, pp. 303–317, 2005.
- [11] M. Mavrovouniotis and S. Yang, "Ant colony optimization with immigrants schemes for dynamic environments," in *Proceedings of the 11th International Conference on Parallel Problem Solving from Nature*, ser. LNCS, vol. 6239. Springer-Verlag, 2010, pp. 371–380.
- [12] —, "A memetic ant colony optimization algorithm for the dynamic travelling salesman problem," *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, vol. 15, no. 7, pp. 1405–1425, 2011.
- [13] —, "Ant colony optimization with immigrants schemes for the dynamic vehicle routing problem," in *EvoApplications2012: Applications of Evolutionary Computation*, ser. LNCS, vol. 7248. Springer-Verlag, 2012, pp. 519–528.
- [14] —, "Adapting the pheromone evaporation rate in dynamic routing problems," in *EvoApplications 2013: Applications of Evolutionary Computation*, ser. LNCS, vol. 7835. Springer-Verlag, 2013, pp. 606–615.
- [15] —, "Ant colony optimization with immigrants schemes for the dynamic travelling salesman problem with traffic factors," *Applied Soft Computing*, vol. 13, no. 10, pp. 4023–4037, 2013.
- [16] —, *Dynamic vehicle routing: A memetic ant colony optimization approach*. Springer-Verlag, 2013, ch. 9, pp. 283–301.
- [17] M. Mavrovouniotis, S. Yang, and X. Yao, "A benchmark generator for dynamic permutation-encoded problems," in *Proceedings of the 12th International Conference on Parallel Problem Solving from Nature*, ser. LNCS, vol. 7492. Springer-Verlag, 2012, pp. 508–517.
- [18] L. Melo, F. Pereira, and E. Costa, "Multi-caste ant colony algorithm for the dynamic traveling salesperson problem," in *Proceedings of the 11th International Conference on Adaptive and Natural Computing Algorithms*, ser. LNCS, M. Tomassini, A. Antonioni, F. Daolio, and P. Buesser, Eds., vol. 7824. Springer-Verlag, 2013, pp. 179–188.

- [19] R. Montemanni, L. M. Gambardella, A. E. Rizzoli, and A. V. Donati, "Ant colony system for a dynamic vehicle routing problem," *Combinatorial Optimization*, vol. 10, pp. 327–343, 2005.
- [20] A. E. Rizzoli, R. Montemanni, E. Lucibello, and L. M. Gambardella, "Ant colony optimization for real-world vehicle routing problems - from theory to applications," *Swarm Intelligence*, vol. 1, no. 2, pp. 135–151, 2007.
- [21] A. Simões and E. Costa, "Chc-based algorithms for the dynamic traveling salesman problem," in *Applications of Evolutionary Computation*, ser. Lecture Notes in Computer Science, C. Chio, S. Cagnoni, C. Cotta, M. Ebner, A. Ekrt, A. Esparcia-Alczar, J. Merelo, F. Neri, M. Preuss, H. Richter, J. Togelius, and G. Yannakakis, Eds. Springer Berlin Heidelberg, 2011, vol. 6624, pp. 354–363.
- [22] S. Yang, "Genetic algorithms with elitism-based immigrants for changing optimization problems," in *Applications of Evolutionary Computing*, ser. Lecture Notes in Computer Science, M. Giacobini, Ed. Springer Berlin Heidelberg, 2007, vol. 4448, pp. 627–636.
- [23] X. Yu, K. Tang, and X. Yao, "Immigrant schemes for evolutionary algorithms in dynamic environments: adapting the replacement rate," *Science China, Information Sciences*, vol. 14, no. 7, pp. 1352–1364, 2011.